UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/776,435	02/11/2004	Kyle Marvin	BEAS-01399US2	2900

23910 7590 03/17/2008

FLIESLER MEYER LLP
650 CALIFORNIA STREET
14TH FLOOR
SAN FRANCISCO, CA 94108

EXAMINER	
VU, TUAN A	
ART UNIT	**PAPER NUMBER**
2193	
MAIL DATE	**DELIVERY MODE**
03/17/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/776,435	MARVIN, KYLE	
	Examiner	Art Unit	
	Tuan A. Vu	2193	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *01 January 1962*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-62* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-62* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *12/21/2007*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This action is responsive to the Applicant's response filed 2/04/08.

As indicated in Applicant's response, claims 1, 7, 17 have been amended, and claims 63-64 canceled. Claims 1-62 are pending in the office action.

Claim Rejections - 35 USC § 101

2. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

3. Claims 1-22 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

> The Federal Circuit has recently applied the practical application test in determining whether the claimed subject matter is statutory under 35 U.S.C. § 101. The practical application test requires that a " useful, concrete, and tangible result" be accomplished. An "abstract idea" when practically applied is eligible for a patent. As a consequence, an invention, which is eligible for patenting under 35 U.S.C. § 101, is in the "useful arts" when it is a machine, manufacture, process or composition of matter, which produces a concrete, tangible, and useful result. The test for practical application is thus to determine whether the claimed invention produces a "useful, concrete and tangible result".

The current focus of the Patent Office in regard to statutory inventions under 35 U.S.C. § 101 for method claims and claims that recite a judicial exception (software) is that the claimed invention recite a practical application. Practical application can be provided by a physical transformation or a useful, concrete and tangible result. The following link on the World Wide Web is for the United States Patent And Trademark Office (USPTO) policy on 35 U.S.C. §101. <http://www.uspto.gov/web/offices/pac/dapp/opla/preognotice/guidelines101_20051026.pdf>

Specifically, claim 1 recites a computer-implemented system comprising a runtime container, a metadata object and runtime architecture, all of the latter being construed as application level software entities or elements to implement a runtime container framework of

Figure 4 in the Specifications. All of which are thus perceived as mere "Functional Descriptive Material" per se, for the claim does not convey that hardware support is included with the system such to execute any software function and realize functionality of the recited elements (see Annex IV, of Guidelines, pg. 34-35). The claim amounts to a system that cannot have embodiment equipped with hardware able to execute the functionality as recited to yield a real-world result (i.e. tangible, concrete and useful result of a practical Application). Further, software listing per se cannot be deemed as belonging to any of the 4 statutory categories of subject matter (apparatus, composition of matter, process, article of manufacture). Claim 1 is non-statutory for recited descriptive software entities per se.

Claims 2-6 fail to remedy to the lack of hardware support hence are likewise rejected.

Claim 7 is a computer-implemented system that comprises 'routing component', 'invocation component', a 'service component', a 'state manager', and 'control component', which are described as mere application software-based functionalities. As such, the claim amounts in a whole, to what is termed as "Functional Descriptive Material' (see 101 Guidelines, Annex IV). For the same reasons as set forth above, claim 7 is rejected for not conveying a hardware support to carry out the above functionality, and is non-statutory for not belonging to any statutory category of subject matter, and for not being able to realize software functionality and yielding tangible, concrete and useful result.

Claim 8-16 fail to remedy to claim 7, hence are rejected for leading to non-statutory subject matter.

Claim 17 also recites a computer-implemented system with elements construed as listed software functionalities, and as set forth above, is rejected for leading to non-statutory subject matter along with dependent claims18-22.

Claim Rejections - 35 USC § 102

4. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

5. Claims 1-18, 21-39, 41-59, 61-62 are rejected under 35 U.S.C. 102(e) as being anticipated by Pace et al, USP: 7,181,731 (hereinafter Pace).

As per claim 1, Pace discloses a system to provide a common runtime container framework, comprising:

a runtime container capable of processing service requests and providing application services (see Fig. 13-15; *EJB container* – col. 40, lines 42-52 – Note: J2EE application dealing with EJB reads on runtime container for processing using CDS/ADS – see Fig. 15A-B);

a metadata object capable of providing metadata on context, state, and/or other information about the data and objects being processed (see col. 39 line 50 to col. 40, line 22; Fig. 2C; Fig. 13 – Note: support for the LD layer by EE 220 as one layer in the EIS reads on providing metadata or descriptors on asset context, logic state, or dynamic content status – see asset types – see col. 41 line 40 to col. 42, line 67); and

a hierarchical architecture (e.g. Fig. 2A and related text; layers – col. 39, line 8 to col. 40, line 22) capable of organizing the runtime container and the metadata object at levels within the hierarchical architecture (see container - Fig. 15B – Note: deployment layer working with analysis of assets to retrieve J2EE beans into a container framework via adapter methodologies to validate descriptor metadata and extend asset by OO package – see Fig. 3, Fig. 13A - reads on organizing of container and metadata at levels within a hierarchical architecture – see Figs. 13, 14, 15).

As per claims 2-3, Pace discloses wherein: the runtime container is extensible (extended environment – Fig. 13A; Fig. 2A) via inheritance mechanisms, which inherit, provide, create and extend services, functionalities and properties of other runtime containers in the hierarchical architecture (e.g. Fig. 3, 13A-B – Note: J2EE beans retrieved from object-oriented hierarchy of assets to deploy container disclose Java inheritance components used to extend functionalities via container framework, adapter and asset discovery services – see Fig. 2; 13-15; col. 85, lines 41 to col. 86, line 17) wherein the metadata object (e.g. Fig. 3, 13) is extensible via inheritance mechanisms, which inherit properties, methods and interfaces of other metadata objects in the hierarchical architecture.

As per claim 4, Pace discloses wherein: the runtime container and the metadata object are organized in duality, wherein a container at one level in a hierarchical architecture is capable of accessing a metadata object at the same level (Adaptor method – Fig. 16 – Note: creating of descriptor for a asset based on EB or SB type reads on one layer of asset per deployment container – see Fig. 15; col. 85, lines 41 to col. 86, line 17) in the hierarchical architecture.

As per claim 5, Pace discloses a well-defined API (col. 36, lines 4-54; Fig. 2A) capable of creating new types of runtime containers, or customizing existing containers with incremental features.

As per claim 6, Pace discloses a well-defined API (refer to claim 5) capable of creating new levels (e.g. extension – Fig. 13B; adaptor - Figs. 17, 18) in the hierarchical architecture for the runtime container and the metadata object (e.g. col. 85, lines 41 to col. 86, line 17 – Note: refer to claim 4 for one layer of asset per deployment package using one adaptor instance).

As per claim 7, Pace discloses a system to provide a common runtime container framework, comprising:

a routing and event handling component capable of communicating (e.g. step 2140A, Fig. 19C) with an invocation component and adapted to be capable of communicating with external clients (Fig. 2B; Fig. 9); said invocation component capable of :

receiving requests from the routing and event handling component (e.g. Fig. 15C; Fig. 18A; Fig. 19C; Fig. 10; Fig. 22, 24 Note: redirect assets distribution reads on routing; and fulfill of asset adapting request reads on return response from requests),

dispatching said requests to a service (e.g. Fig. 15C; Fig. 18A; Fig. 19C; Fig. 10; Fig. 22, 24 Note: redirect assets distribution reads on routing or dispatching) component within a runtime container (Fig. 15), and

managing the returned responses (Fig. 2B; Fig. 15C; Fig. 18A; Fig. 19C; Fig. 10; Fig. 22, 24 Note: fulfilling of asset adapting request reads on return response from requests);

at least one said service component within the runtime container capable of processing requests from the invocation component and producing responses back to the invocation

component (e.g. Fig. 15; col. 85, lines 41 to col. 86, line 17; deliver and deploy – col. 86, lines 10-17; Fig. 15B-C);

a state manager capable of obtaining state information from a nonvolatile storage and providing such information (e.g. Fig. 13, 14 – Note: database asset and discovery of asset using adaptor service reads on non-volatile get state information in regard to support from container deployment framework in regard to a request or invocation; see *state of the database* – col. 88, lines 4-13) to the invocation component and the runtime container (e.g. Fig. 15B-C);

a context manager capable of obtaining information from a metadata object and providing such information to the invocation component and the runtime container(see Fig. 13-14); and

a control component capable of communicating with the runtime container and adapted to be capable of communicating with external services (Fig. 1D; Fig. 10; EIS database Fig. 15B, 15C; RMI Fig .16) .

As per claim 8, Pace discloses the routing and event-handling component is capable of communicating with the invocation component using a uniform or standardized protocol (Fig. 9-10 Note: COM, RMI, HTTP or TCP/IP of network reads on uniform protocol – e.g. col. 36, lines 4-39).

As per claim 9, Pace discloses wherein the invocation component can be event-driven (col. 87, lines 5-19), wherein event delivery to a component and event generation from a component within the runtime container can be synchronous (e.g. synchronization - col. 87, lines 45 to col. 88, lines 14) or asynchronous.

As per claim 10, Pace discloses wherein: the service component can be created in the form of Java Beans (e.g. J2EE col. 87, lines 45 to col. 88, line 2).

As per claim 11, Pace discloses wherein: the service component is capable of performing either pre-processing requests (Fig. 13A) or post-processing of responses sent to or returned from the component (e.g. Fig. 14A-B; col. 85, lines 41 to col. 86, line 17).

As per claim 12, Pace discloses a simplified component abstraction capable of transparently mapping the service component to a more complex set of components (adaptor – Fig. 14, Fig 16) to generate and deploy applications under runtime environment.

As per claim 13, Pace discloses a common configuration model (CORBA – col. 61, lines 48-65; CORBA -Fig. 10) capable of specifying the service component declaratively and programmatically, as well as providing a model for declarative configuration override of the component at application deployment time (Note: J2EE container creation reads on programmatic overriding over specification of DCOM or a Corba model).

As per claim 14, Pace discloses wherein: the state manager is capable of locating, managing and persisting state information, wherein the physical mechanism of doing so is transparent (Fig. 16 – Note: invocation via RMI is transparent to requesting client).

As per claim 15, Pace discloses wherein: the context manager is capable of exposing component-level application services using the context information (col. 65, line 21 to col. 66, line 46; Fig. 13B Note: database context reads on context information obtained from exposed information in markup – see Fig. 4, 5, 12).

As per claim 16, Pace discloses wherein: the control component is capable of providing a simplified and common interaction model to communicate with the external services (e.g. col. 36, lines 12-54; Fig. 2b; Fig. 10).

As per claim 17, Pace discloses system to provide a common runtime container, comprising:

at least one servlet (col. 36, lines 12-54; Fig. 2b; Fig. 10 – Note: services provided via a single API framework reads on servlet) capable of managing communications (e.g. step 2140A, Fig. 19C) between the runtime container and external entities (e.g. Fig. 2B, 9; Fig. 15B, 15, 16) using common or uniform protocols (refer to claim 8);

at least one listener capable of monitoring incoming communication at the servlet (see Fig. 2, 9, 10);

a first dispatcher component capable of :

communicating with one or more servlets, determining which components to invoke, dispatching requests requiring asynchronous processing to a queue (e.g. Fig. 13C-D), and

dispatching requests requiring synchronous processing directly to a stateful or a stateless component (refer to claim 8); said queue capable of storing asynchronous requests (Note: network of Fig. 9 reads on asynchronous packet arrival);

a second dispatcher component capable of

receiving requests from the queue, determining which components to invoke, and dispatching requests (Fig. 16) requiring synchronous processing directly to a stateful or a stateless component (Fig. 13-14 - Note: objects being parsed prior to discovery via state database reads on stateless); at least one said stateless component capable of processing stateless requests; and at least one said stateful component capable of processing stateful requests(Note: this is taught by the very nature of web services and request fulfilling – see Fig .9, 10 – as well as the re-routing of claim 8).

As per claim 18, Pace discloses wherein: the servlet is capable of communicating in TCP/IP, HTTP, SOAP, XML (e.g. Fig. 9-10; HTTP – col. 3; top; col. 37, lines 10-32; Fig. 28E; col. message middleware col 48 lines 19-31), and other application-specific protocols (see above).

As per claim 21, Pace discloses wherein the stateless component (e.g. Fig. 13 – Note: web markup language reads on stateless) is capable of at least one of the following:

deriving context information from the metadata; containing an arbitrary amount of code for processing logic; calling other stateless components within the container (col. 85, lines 41 to col. 86, line 17); and utilizing synchronous or asynchronous controls to communicate with external services (see Fig. 14-15).

As per claim 22, Pace discloses wherein: the stateful component is capable of at least one of the following:

deriving context information from the metadata (refer to claim 21); retrieving state information from nonvolatile storage through a state management component; containing an arbitrary amount of code (e.g. col. 36, lines 12-54 -Note: beans container reads on fetching a pool of beans code arbitrary stored for COM or Corba reuse) for processing logic; calling other stateless or stateful components within the container (see Fig. 15B; Fig. 16); and utilizing synchronous or asynchronous controls to communicate with external services (e.g. Fig. 10; Fig. 16 - refer to claim 21 – Note: in view of the nature of the client/server network where data arrival can be stateful or stateless – HTTP data events back as response reads on stateless and return from database queries reads on stateful).

As per claim 23, Pace discloses a method to provide a common runtime container framework, comprising:

processing service requests and providing application services (e.g. Fig. 2C; Fig. 9-10) via a a runtime container (e.g. Fig. 15B); providing metadata on context, state, and/or other information (Figs. 14) about the data and objects being processed via a metadata object (e.g. Fig. 13A); and

organizing the runtime container and the metadata object at levels within a hierarchical architecture (e.g. Fig. 2A and related text; *layers* – col. 39, line 8 to col. 40, line 22).

As per claims 24-28, refer to claims 2-6, respectively.

As per claim 29, Pace discloses a method to provide a common runtime container framework, comprising:

communicating with an invocation component and external clients via a routing and event handling component;

receiving requests from the routing and event handling component, dispatching said requests to a service component within a runtime container, and

managing the returned responses via the invocation component; processing requests from the invocation component and producing responses back to the invocation component via the service component within the runtime container;

obtaining state information from a nonvolatile storage and providing such information to the invocation component and the runtime container;

obtaining information from a metadata object and providing such information to the invocation component and the runtime container; and

communicating with the runtime container and external services;

all of which functionalities having been addressed in claim 7.

As per claims 30-37, refer to the rejection of claims 8, 10-16 respectively.

As per claim 38, Pace discloses a method to provide a common runtime container, comprising:

managing communications between the runtime container and external entities using common or uniform protocols via at least one servlet;

monitoring incoming communication at the servlet;

communicating with one or more servlets, determining which components to invoke, dispatching requests requiring asynchronous processing to a queue, and dispatching requests requiring synchronous processing directly to a stateful or a stateless component;

storing asynchronous requests via the queue; receiving requests from the queue, determining which components to invoke, and dispatching requests requiring synchronous processing directly to a stateful or a stateless component;

processing stateless requests via at least one stateless component; and processing stateful requests via at least one stateful component;

all of which functionalities having addressed in claim 17.

As per claims 39, 41-42, refer to the rejection of claims 18, 21-22 respectively.

As per claim 43, Pace discloses a machine-readable medium having instructions stored thereon that when executed by a processor cause a system to:

process service requests and provide application services via a a runtime container;

provide metadata on context, state, and/or other information about the data and objects being processed via a metadata object; and

organize the runtime container and the metadata object at levels within a hierarchical architecture;

all of which functionalities having been addressed in claim 23.

As per claims 44-48, refer to the rejection of claims 24-28 respectively.

As per claim 49, Pace discloses a machine readable medium having instructions stored thereon that when executed by a processor cause a system to:

communicate with an invocation component and external clients via a routing and event handling component;

receive requests from the routing and event handling component, dispatch said requests to a service component within a runtime container, and manage the returned responses via the invocation component;

processing requests from the invocation component and producing responses back to the invocation component via the service component within the runtime container;

obtaining state information from a nonvolatile storage and providing such information to the invocation component and the runtime container;

obtaining information from a metadata object and providing such information to the invocation component and the runtime container; and

communicating with the runtime container and external services;

all of which having been addressed in claim 29.

As per claims 50-57, refer to the rejection of claims 30-37 respectively.

As per claim 58, Pace discloses a machine readable medium having instructions stored thereon that when executed by a processor cause a system to:

manage communications between the runtime container and external entities using common or uniform protocols via at least one servlet;

monitor incoming communication at the servlet;

communicate with one or more servlets, determine which components to invoke, dispatch requests requiring asynchronous processing to a queue, and dispatch requests requiring synchronous processing directly to a stateful or a stateless component;

store asynchronous requests via the queue; receive requests from the queue, determine which components to invoke, and dispatch requests requiring synchronous processing directly to a stateful or a stateless component; process stateless requests via at least one stateless component; and

process stateful requests via at least one said stateful component;

all of which having been addressed in claim 38.

As per claims 59, 61-62, refer to claims 39, 41-42.

Claim Rejections - 35 USC § 103

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

7. Claims 19-20, 40, 60 are rejected under 35 U.S.C. 103(a) as being unpatentable over by Pace et al, USP: 7,181,731.

As per claim 19, Pace does not disclose the first or second dispatcher can be implemented using Java programming language in the form of Java Beans. But Pace's JEEE-based environment for assembling beans using adapter service and queries to database (see Fig. 15A-C) also provide bytecodes used in effectuating such remote calls (see col 94, line 59 to col. 95, line 4). It would have been obvious for one skill in the art at the time the invention was made to implement in J2EE the remote calls to dispatch a query for additional asset discovery via using a database service call as suggested above, because the beans is highly portable and dynamically for event such as those required during the assets discovery by Pace, and applying J2EE bean to implement the dispatching invocation would alleviate extraneous integration of heterogeneous programming resources.

As per claim 20, the limitation as to the stateless or stateful component can be implemented using Java programming language in the form of Java Beans is deemed to fall under the ambit of using beans for effectuating dispatching invocation; hence would have been obvious as set forth in claim 19.

As per claim 40, the limitation as to implement stateful and stateless component in Java Beans has been addressed using the rationale of claims 19-20.

As per claim 60, the limitation as to implement stateful and stateless component in Java Beans has been addressed using the rationale of claims 19-20.

Response to Arguments

8. Applicant's arguments filed 2/04/08 have been fully considered but they are not persuasive. Following are the Examiner's observation in regard thereto.

USC § 102(e) Rejection:

(A) Applicant has submitted that (for claim 1) Pace's asset definition data structure and multiple layers for an asset are not *hierarchical architecture* capable of organizing runtime container and metadata objects *at levels* within the hierarchical architecture as required in claim 1(Appl. Rmrks pg. 15, bottom, pg. 16 top). The Rejection has addressed the hierarchical architecture in terms of architecture functionality which is phrased as 'organizing runtime container and metadata objects at levels within the … architecture', and this approach has been applied to addressing other claimed entities like 'runtime container' or 'metadata'. Accordingly, 'container runtime' has been analogized to one layer among Pace's multi-layer servicing framework or EIS (i.e. hierarchical architecture); that is, such application is (see Fig. 15B) is viewed as support for **processing** EJB components needed for the extensible layer application (see *Extended Environment* -- Fig. 2A-B) which in turn, supports a LD layer (see Fig. 2A) via interfacing (see EE, AI, BE, cols.39-40) with the BE database layer to obtain **descriptors** for assets, thus enabling the packaging of these based on the context of assets required for the LD runnable objects; and metadata has been analogized to descriptor information. The organizing of assets based on descriptors and validating of types and dependencies using the EE 220 - including the layer to process EJB components -- (see Fig. 2C, Fig. 15) and BD layer via rearranging dependencies (e.g. based on runtime context -- Dynamic Content) and categorizing of types (see col. 41-42). This Extensible Environment layer in conjunction with the assets deployment based on adapter services and other layers has been analogized as the organizing in

Pace's EIS with discovery and re-adapting of existing assets via support from EE service and EJB processing service) as claimed, in order to customize runnable packages (see Fig. 13) for the runtime as required by the Logic/Data layer 210 (see Fig. 2B). "At levels within the hierarchical architecture' has been treated as any one layer application as exemplified above, absent specific language from the claim that would describe further what 'at levels' entails; accordingly, Pace has disclosed 'organizing the runtime container and the metadata object at levels within the hierarchical architecture'. The claim language for lack of further specifics cannot be viewed as sufficiently detailed as to preclude Pace's EIS framework from fulfilling the language of claim 1. Applicant's arguments fail to comply with 37 CFR 1.111(b) because they amount to a general allegation that the claims define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the reference.

(B) Applicant has submitted that (for claim 7) Pace Fig. 15 cannot anticipate "at least one said service component within the runtime container … responses back to the invocation component' (Appl. Rmrks pg. 16, middle). The rejection has addressed runtime container as application under the LD layer for addressing EJB when data returned from the BD via requests from the EE layer in order to provide dependency and metadata for the adapter/discovery service of the EIS. And all the receiving/returning of such data including dispatching for more or more redirecting via distribution for extending the amount of valid assets proper for some logic has been identified in the Office Action, such that when fulfilled the complete asset (or runnable assets package) would be returned to the requesting client. The claim language does not specifically depict how each of the *request* processing instance and *response* generating instance consists of in order to preclude the asset discovery and delivery by Pace's container application

and multi-layer EIS from fulfilling the above 'processing requests' and 'producing responses' limitation. The argument is not sufficient to overcome the rejection.

(C) Applicant has submitted that (for claim 17) the Office action provides no teaching about communications between the runtime container and external entities (Appl. Rmrks pg. 16, bottom). The rejection has pointed to Pace Figures whereby descriptor retrieved (from external EIS database) through the interfaces of communication between the CDS/ADS and the DIS container processing application (Fig. 15B-C) are processed and assembled as validated deploy data for deployable package via the adapter service of Fig. 16. The claimed limitation for lack of further details has been deemed fulfilled by Pace. Applicant's arguments fail to comply with 37 CFR 1.111(b) because they amount to a general allegation that the claims define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the reference.

(D) Applicant has submitted that the Office Action does not appear to disclose Pace in terms of first dispatcher synchronous communication and second dispatcher's asynchronous communication (Appl. Rmrks pg. 16 to pg. 17). This remark appears to be a short cut leaping to a foregone conclusion that the so-recited first and second *dispatchers* are patentable subject matter. However, the remark does not explain how the cited parts of Pace fails to match the very specific of the claim; rather the Applicant merely provides a assertion for patentability without **clearly proving how** the part provided by the Office action would distinguish with either *dispatcher* as recited. The rejection has pointed to request for discovery of assets and the cited Figures for showing out-going requests are deemed to map synchronous requests (for first dispatcher). On the other hand, packets or received data arriving and queued can be seen as

second dispatcher asynchronous method; and such queueing has been cited in the rejection. The claim language regarding the above first and second dispatcher has been given weight in terms of synchronous and non-synchronous queuing; that is, the above allegation of patentability is not deemed sufficient to overcome the rejection, for the Applicant fails to put forth a proper prima facie type of argument compliant to CFR 1.111b as set forth above.

(E) In all, the claims including those rejected under USC 103(a), stand rejected as set forth in the Office Action.

Conclusion

9. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tuan A Vu whose telephone number is (571) 272-3735. The examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Lewis Bullock can be reached on (571)272-3759.

The fax phone number for the organization where this application or proceeding is assigned is (571) 273-3735 (for non-official correspondence - please consult Examiner before using) or 571-273-8300 (for official correspondence) or redirected to customer service at 571-272-3609.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100.
Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Tuan A Vu/

Primary Examiner, Art Unit 2193

March 10, 2008